FLY SAFE, BUT FLY

by kenneth j. szalai

Risk management process is a front-burner topic in all aircraft and spacecraft communities. A former colleague of mine noted that a zero mishap rate in experimental flight research can only be assured by padlocking the hangar doors. However, achievement of the objectives of a flight program still requires accomplishment of the mission—flying.

I learned a lesson about managing risk from a great project manager—Calvin R. Jarvis. The Phase I program adapted the Apollo Guidance System to the F-8 airplane, and in 1972 it flew without any mechanical pilot controls, becoming the first aircraft to fly with a digital computer in full control.

I got to work for Cal Jarvis on the F-8 Digital Fly-by-Wire (DFBW) project. The Phase I program adapted the Apollo Guidance System to an airplane, In 1972 it flew without any mechanical back-up system, becoming the first digital fly-by-wire aircraft. Its safety depended on its full-time, full-authority primary DFBW control system.

Cal managed both high-risk pioneering programs with great success. He was a master Project Manager, always keeping cost, schedule and technical performance in balance, but never trading them for safety.

The Crisis

I was chief engineer and software manager of the second phase of the program, where three digital computers were configured as the first fault-tolerant airplane DFBW system. The Draper Laboratory developed the flight software. IBM supplied the flight computers.

The F-8 Digital Fly-by-Wire (DFBW) in flight.

The F-8 DFBW aircraft had flown about a dozen flights, and was making good progress. A systems engineer called and told me that the preflight self-test had failed while preparing for the next day's flight.

Early in the program we had encountered a few self-test problems in the control surface tests, which had very small tolerances, so I assumed it was a tolerance problem.

While troubleshooting the self-test failure on the airplane, however, I froze and my heart sank as I realized the problem was far worse than some self-test tolerance setting. I discovered that a half-dozen instructions in the computer's flight software did not match the program listing! I could tell the paper listing was correct — so the flight computer had contaminated instructions.

This was impossible, unbelievable. There was no immediate explanation. One thing was certain, the airplane and the program had to be grounded until we could figure out what had happened.

The F-8 Digital Fly-by-Wire (DFBW) in flight.

I called Cal Jarvis immediately and told him about the problem. He asked quietly, "Does this mean the flight tomorrow is off?" "Cal," I said, "This is a catastrophe! I have no idea what is wrong. This is a spear through the heart of the program."

"How about flying Tuesday or Wednesday next week?" Cal asked. "You just don't understand," I said. "This is a monumental disaster." Cal wasn't listening to my doomsday remarks. He asked me to document the problem and work with Draper and IBM to find out what happened.

Problem Found, but the Crisis Remains

The Draper Laboratory and IBM identified the cause of the problem the next day. An error in the "Assembler" software was found that could produce a contaminated computer load tape while correctly producing the software listing.

We verified that the paper listing was correct. This was to be a key finding. We also established that prior flight tapes had not been contaminated. We informed the Space Shuttle program about this problem since they were using the same computer and Assembler software.

I told Cal that IBM was fixing the Assembler flaw and we were developing a new check process that would verify future tapes to be correct. We would then reassemble the flight code, generate a new load tape, load the computers and carry out the preflight tests. I estimated it would take a couple of weeks.

Risk Management

Cal noted that the F-8 schedule was critical, with upcoming flight tests of some Space Shuttle software. On Thursday he asked, "Is next week out of the question for a flight?" By this time my patience was wearing thin. I was only thinking of safety, my primary focus. Cal was concerned about safety, too, but he was also thinking about flying safely.

He asked me how many memory locations had been contaminated. I said that the handful of instructions in the self-test program was the only thing we'd found so far, but that there were about 25,000 instructions and data words that we hadn't checked yet.

Cal then asked the key question, "If we could prove that these were the only contaminated memory locations, and we corrected them, would we be able to fly next week?" That was a good question—could we fix the problem expeditiously? I looked at the pictures on his wall, of the other experimental aircraft he had worked on, which all flew successfully. I decided to think about his challenge.

Risk Elimination

I subdued my emotional response, and started to focus on the technical issues. We didn't have a means to automatically check the computer memory against the accurate printed listing. The listing took up 250 big pages.

I laughed to myself and thought, "How long would it take to manually check a computer memory dump against the listing?

Let's see, there are 25,000 memory locations. If we had five teams of engineers, and they could read aloud and verify one memory location every 10 seconds, five teams could verify 30 memory locations a minute. That would take about 14 hours.

I proposed this to Cal, and he smiled and said, "How about flying next Wednesday?" I said we could do it. We got a few more than five teams together, alternated the reader and verifier every couple of pages or so, and added breaks. We finished by Friday afternoon, and did not find any other errors. I guess sometimes pioneering work needs solutions rather than elegance.

We had a process for patching the software on site, and so we manufactured and verified a corrected load tape. The self-test was run and passed several times. I signed the software release document as the Software Manager (with more humility this time) and developed the technical briefing for management.

I presented the technical findings and explained the carefully controlled manual checking process. Cal offered another explanation for flight readiness. "We all agree that the intended software load, as represented by the listing, was qualified before the contamination, and we had many successful flights. Therefore correcting the six incorrect computer words to their original value means the patched software load is also qualified."

After some hard questions—like "Why didn't you anticipate this?"—senior management approved the flight. We flew on Wednesday, as Cal had asked.

We also worked with Draper Lab and IBM to develop a 100% closed loop check process of the load tape and listing against the original machine code produced in the mainframe.

Years Later

This event happened more than 25 years ago. The F-8 DFBW completed a long and safe flight research program.

I learned from Cal Jarvis that the role of the Project Manager is to complete the project successfully. Cal Jarvis never lost sight of that, but he never cut any corners or sacrificed safety for schedule. I realized Cal Jarvis's success on the F-8 DFBW was not just good luck.

Cal Jarvis went on to manage several more projects, including the joint US-Russian Tu-144 supersonic flight research project, and eventually ran the entire Dryden Flight Projects organization for several years.

The bottom line is that it is possible to conduct high-risk missions safely, but it takes intense effort and an open, communicating organization. Today, you can visit Dryden and see the F-8 DFBW, the X-29, and several other experimental aircraft on display. This is a testimony to the ability to fly high-risk programs safely.

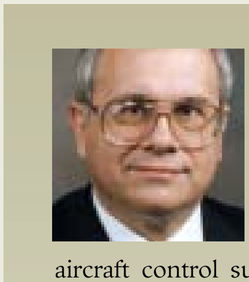
It had been instilled in me that the objective of an important endeavor, like experimental flight research and test, is to fly safely. Padlocking the hangar doors will eliminate flying accidents, but will not advance the cause of flying. •

Lessons

- Effective managers keep their cool when unexpected situations arise. This enables them to calmly see the big picture while zeroing in on the specific problem.
- Project success requires a keen focus on results, even if the processes and solutions employed to get them are not so elegant.

Question

To what extent is the project leader's attitude towards a setback reflected in the attitude of the team?



KENNETH J. SZALAI served as Director of the NASA Dryden Flight Research Center from 1990–1998, retiring from NASA after 34 years of service. He is currently an aerospace and management consultant in the United States and in Europe.

Digital Fly-by-Wire, or DFBW, is a digital electronic flight control technology. A digital computer receives pilot maneuver commands and rate, altitude, and acceleration information from sensors. The computer uses this information to direct the hydraulic actuators to move the aircraft control surfaces in such a way that the aircraft maneuvers according to the pilot's commands. All this information is electronic, and is carried by wires, hence "fly-by-wire". The cables, pulleys, and other mechanical devices formerly used to connect the pilot stick controls to the surfaces or hydraulics are removed in a pure fly-by-wire system.

The digital computer software also provides powerful artificial stabilization of the vehicle so that the aircraft designer is no longer bound by traditional design constraints. DFBW enabled the first moon landing. The NASA Dryden F-8 DFBW program was the first to achieve digital fly-by-wire control for an aircraft, in May 1972. Now virtually all military aircraft and modern airliners use DFBW control.